

REMARKS

Rejection of claim 1 over *Masiu*

Claim 1 stands rejected as rejected over *Masiu* (US Pat. 5,179,632). Claim 1 requires in part

processing the rules base to form a data structure in a computing system, each rule being associated with a corresponding portion of the data structure, each corresponding portion representing the condition for the rule and including storage locations for holding values of the condition elements of the conditions for said rule.

The prior Office Action also rejected the claim over *Masiu*, citing different portions of the reference in support of the rejection. In response, the Applicant argued in part:

... Masui does not disclose or suggest that a node holds the values of a rule's condition elements. For example, in Masui each rule could link to multiple nodes and therefore the nodes that may hold the values of the condition elements are not each associated with a particular rule. Therefore, Masui does not designate to each rule having a corresponding portion of the data structure for the rule's condition.

Additionally, even if Masui does teach that each rule has a corresponding portion of a data structure and that "the data structure includes ... storage locations," which the applicant does not concede is true, Masui neither discloses nor suggests that the corresponding portion of the data structure includes the storage locations, because Masui fails to specify where in the data structure the storage locations are situated. Therefore, Masui neither discloses nor suggests that the storage locations are situated in a portion of the data structure corresponding to a rule.

In response, the present Office Action takes the position that the recited "corresponding portion of the data structure" that represents a condition of a rule is the network (or apparently a corresponding portion of the network) through which information is processed. The Office Action further takes the position that nodes of the network in *Masui* hold values of a rule's condition elements by virtue of condition clauses being stored in each node. The Applicant disagrees with these positions.

The Rete Algorithm

The Office Action relies on a portion of the Background of *Masui* (col.1, lines 24-54), which describes the well-known Rete algorithm as the basis for anticipation of claim 1. This portion of *Masui* reads in part:

... The current data are supplied to the network to check if the data satisfy the condition clauses stored in each node. If satisfied, the data are supplied to the next node, whereas if not, the process is terminated. The above processes are repeated. If the process reaches the terminal node, it is judged that the condition indicated by the condition name (rule name) described in the terminal node has been satisfied. This method is advantageous in its fast processing.

...

The present application also describes the Rete algorithm as follows:

[06] ... The Rete algorithm uses a data flow network to represent the conditions of the rules. The network has two parts, one part performs the tests required to evaluate the individual condition elements, and a second part combines the condition elements to form the overall conditions for the rules. The outputs of the first part (and the input to the second part) are a set of memories each associated with a different condition element for holding the set of facts that satisfy (or may satisfy for some values of the variables) that condition element. The second part includes storage elements associated with different combinations of condition elements, each identifying the possible corresponding combinations of facts that satisfy that combination of condition elements. Some of these storage elements are associated with overall conditions that particular rule must satisfy, and the presence of any combination of facts in these storage elements indicate that the corresponding rules are applicable given those facts.

Turning to the Rete paper ("Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem"), which is incorporated into *Masui* by reference (col. 1, lines 13-16), we see that the content of the network is specified as follows:

The algorithm that will be presented here, the Rete Match Algorithm, can be described as an indexing scheme that does not require the interpretive step. The indexing function is represented as a network of simple feature recognizers. This representation is

related to the graph representations for so-called structured patterns. (p. 21)

A pattern matcher can avoid iterating over the elements in working memory by storing information between cycles. The step that can require iteration is determining whether a given pattern matches any of the working memory elements. The simplest interpreters determine this by comparing the pattern to the elements one by one. The iteration can be avoided by **storing, with each pattern, a list of the elements that it matches.** The lists are updated when working memory changes. **When an element enters working memory, the interpreter finds all the patterns that match it and adds it to their lists.** When an element leaves working memory, the interpreter again finds all the patterns that match it and deletes it from their lists. (p. 21-22, emphasis added)

2.2.3. Saving information in the network

As explained above, the black box must maintain state information because it must know what is in working memory. In simple Rete networks all such state is stored by the two-input nodes. **Each two-input node contains two lists called its left and right memories. The left memory holds copies of the tokens that arrived at its left input, and the right memory holds copies of the tokens that arrived at its right input. The tokens are stored as long as they are useful.** The next section explains how the nodes determine when the tokens are no longer useful. (p. 25, emphasis added)

It is therefore clear from these descriptions of the Rete algorithm that there are indeed nodes in a network, and that some of the nodes each correspond to a different condition element and other nodes each correspond to a different rule. However, it is also clear that what is stored at the nodes are “tokens”, which are also known as “work elements” or “facts.”

The Rete algorithm is not within the scope of claim 1

Turning again to the language of claim 1, and the Office Action's apparent position that the recited “corresponding portion [of the data structure with which each rule is associate]” matches a portion of the Rete network that includes a rule's node and the nodes for the condition elements of the rule, then the recited “storage locations for holding values of the condition elements of the conditions for said rule” would have to be found at these nodes. But they are

not. A condition element node in Rete has tokens (or combinations of tokens) that match the condition element. On the other hand, the specification of the present application makes clear that the "value of a condition element" is an indicator of whether the condition element is matched. For example, "[a] condition element is evaluated to be true for a fact if that fact includes attribute/value pairs that together determine that the overall logical function is true," that is, the condition element evaluates to a value that is true or false. There is no storage in the *Masui* network for such values.

The Applicant therefore emphatically disagrees that the Rete algorithm referred to by *Masui* in its background, or the variants of that algorithm described elsewhere in *Masui*, disclose or suggest the form of data structure recited in claim 1.

The remaining claims are patentable over *Masui* for at least the reasons set forth above for claim 1.

Rejections under 35 USC 101

The Office Action has maintained its rejection of the claims as being directed to non-statutory subject matter. The Action states "claims 1-11 are broad in concept. Specifically, in claims 1-11, the concept of rules applies to the entire domain or rules and thereby preempts the abstraction of rules."

This statement appears to utterly ignore the limitations recited in claims. For example, it is very clear as argued above that claim 1 does not include in its scope the previously known Rete algorithm. There is nothing abstract or intangible about the limitations of the claim – "processing the rules base to form a data structure in a computing system, each rule being associated with a corresponding portion of the data structure, each corresponding portion representing the condition for the rule and including storage locations for holding values of the condition elements of the conditions for said rule." This step is tangible in its arrangement of the data structure in a computing system. Furthermore the claimed subject matter is useful as a basis for efficient processing of the rules.

In response to the Applicant's argument in response to the prior rejection under 35 USC 101, the Office Action states "claim set 1-11 attempts to patent 'every substantial application'

regarding rule processing." If this is meant to state that automated rules processing systems are not patentable outside the context of a domain of rules being processed, then the Applicant most emphatically disagrees. By the same rational, instruction processing systems (e.g., computer processors) would not be patentable outside the context of a domain of instructions (e.g., programs) executed on the processing system.

The claims are not directed to abstract ideas and do not preempt the abstraction of rules. The claims meet all the requirements of 35 USC 101.

Please apply \$60.00 for the Petition for Extension of Time fee, and any other charges or credits, to deposit account 06-1050, referencing Attorney Docket No. 16887-002001.

Respectfully submitted,

Date: Jan. 29, 2007

J. Robin Rohlicek
J. Robin Rohlicek, J.D., Ph.D.
Reg. No. 43,349

Fish & Richardson P.C.
225 Franklin Street
Boston, MA 02110
Telephone: (617) 542-5070
Facsimile: (617) 542-8906